

[Inicio](#) | [Fundamentos](#) | [Instalación](#) | [Añadir Elementos](#) | [Encadenar Métodos](#) | [Asociar Datos](#) | [Usando sus Datos](#) | [Desplegar DIVs](#) | [La Función data\(\)](#) | [Introducción a SVG](#) | [Despliegue de SVG](#) | [Tipos de Datos](#) | [Diagrama de Barras](#) | [Diagrama de Dispersión](#) | [Escala](#)s | [Ejes](#) |

Encadenar Métodos

D3 utiliza una técnica denominada sintáxis de cadena, similar a la de jQuery. Mediante el encadenamiento de métodos separados por puntos, es posible ejecutar varias acciones en una misma línea de código. Esto puede ser rápido y fácil, pero es importante entender cómo funciona, para ahorrarse horas de dolores de cabeza depurando el código.

A propósito, *funciones* y *métodos* son dos palabras diferentes para describir el mismo concepto: una unidad de código que acepta un argumento como entrada, lleva a cabo alguna acción, y devuelve algún dato de salida.

Revisemos nuevamente la primera línea de código en D3. [Página de ejemplo](#).

```
d3.select("body").append("p").text("New paragraph!");
```

Esto puede parecer algo desordenado, especialmente cuando se está comenzando a programar. Lo primero que toca saber es que JavaScript, al igual que HTML, ignora el espacio en blanco y los cambios de línea, por consiguiente se puede incluir cada uno de los métodos en su propia línea, lo cual es más legible.

```
d3.select("body")
  .append("p")
  .text("New paragraph!");
```

Cada quién tiene su propio estilo de programar, por consiguiente puede utilizar diferentes espacios de indentación (sangría), cambios de línea, o espacio en blanco de acuerdo con sus preferencias.

Un Enlace a la Vez

Veamos las diferentes partes del código anterior.

d3 - referencia al objeto D3, de tal forma que se pueda acceder a sus métodos.

.select("body") - Al darle a **select()** un selector de CSS como entrada, éste devolverá una referencia al primer elemento del DOM que coincida. (Se recomienda usar **selectAll()** cuando se necesite más de un elemento). En este caso, solo se necesita la etiqueta **body**, por eso se pasa la referencia a **body** al siguiente método de la cadena.

La Transferencia

Muchos pero no todos los métodos devuelven una selección (realmente la referencia a la selección), lo cual permite aplicar la técnica tan útil que es el encadenamiento de métodos. Usualmente, un método devuelve una referencia al objeto que alteró, pero no siempre.

Para recordar: Cuando se encadenan métodos el orden importa. El tipo de salida (output) de un método debe coincidir con el tipo de entrada (input) del siguiente método de la cadena. Si las entradas y salidas de métodos adyacentes no son compatibles, no ocurrirá la transferencia de información.

Esquema sin Encadenamientos

El código del ejemplo también se puede escribir sin la sintaxis de encadenamiento.

```
var body = d3.select("body");  
var p = body.append("p");  
p.text("New paragraph!");
```

Realmente mucho más confuso! Sin embargo, abrá veces cuando será necesario romper la ca

Siguiente: Asociar Datos →

Este tutorial fue traducido por **Gabriel Coch**

Todo el contenido fue desarrollado por y le pertenece a **Scott Murray**